

基于 Leader-Follower 的分队队形控制寻径算法研究 *

陈逸, 高岩[†], 王长波

(华东师范大学 计算机科学与软件工程学院, 上海 200062)

摘要: 在战场环境中, 战术分队的队形在面对复杂静态或动态障碍物难以较好地保持, 针对此问题, 提出了基于 Leader-Follower 算法的改进队形控制方法。在 Leader 寻径阶段, 通过在战场导航网格中应用两阶段路径搜索方法, 先使用 A*算法寻找由三角形通道和可利用地物组成的路径, 再使用改进的 Funnel 算法在考虑队形规模的约束条件下对路径作平滑处理。在 Follower 跟随阶段中, 通过采用 morphing 技术, 产生在复杂障碍约束下平滑的中间约束队形序列, 并结合提出的队形弹簧模型, 局部修正并控制 Follower 每一时刻的速度。为解决面对动态障碍的避碰问题, 基于相对速度障碍法, 并加入速度协同控制, 避免队形在避碰过程中失效。最后通过实验表明了该方法的有效性。

关键词: 队形控制; 分队寻径; 导航网格; 相对速度障碍; 队形弹簧; 变形

中图分类号: TP391.9 **doi:** 10.3969/j.issn.1001-3695.2017.06.0659

Research on team formation control path finding algorithm based on Leader-Follower

Chen Yi, Gao Yan[†], Wang Chang-Bo

(School of Computer Science & Software Engineering, East China Normal University, Shanghai 20062, China)

Abstract: In the battlefield environment, the tactical unit formation is difficult to maintain when facing complex static or dynamic obstacles. For this problem, this paper proposed an improved formation control method based on Leader-Follower algorithm. In the Leader routing stage, by applying the two-stage path search method in the battlefield navigation mesh, this paper used the A* algorithm to find the path composed of the triangular channel and the available objects, and then used the improved Funnel algorithm to smooth the path under the constraints of the scale of unit. In the Follower following stage, this paper used morphing technique to produce smooth intermediate constraint formation sequences under complex barrier constraints and combine the proposed formation spring model to locally correct and control the speed of each follower. In order to solve the problem of collision avoidance in the face of dynamic obstacle, this paper used reciprocal velocity obstacle method, which joined the speed co-control, to avoid the formation failure in the collision avoidance process. Finally, this paper proved the effectiveness of the method by experiments.

Key Words: formation control; unit path finding; navigation mesh; reciprocal velocity obstacle; formation spring; morphing

0 引言

在作战仿真领域, 战术分队的队形控制是一个重要问题, 分队在三维战场环境中寻径的同时, 还需要保持相应的几何阵型。目前常用的队形控制方法有基于行为法、人工势场法、基于图论法以及 Leader-Follower 法。基于行为法需要对 agent 行为建模, 通过感知当前环境选择前进、避障、维护队形等行为^[1], 但存在复杂性问题。人工势场法通过引入物理学的势场, 将 agent 与环境的约束由人工势场和势场函数表示, 通过势场控制 agent 位置^[2], 但存在局部极值点问题^[3]。基于图论法用控制图描述 agent 的队形形状, 并利用控制理论调整队形的姿态^[4], 缺点是实现复杂, 一般只用于仿真研究^[5]。而 Leader-Follower 法

选择其中一个 agent 作为队形的 Leader, 其余的作为 Follower 并通过控制其与 Leader 的距离和角度来形成队形^[6], 将复杂的队形控制问题转变成 Leader 的寻径和 Follower 对 Leader 的跟随问题, 控制器的设计相对简单并比较符合作战的真实情况, 适合作为战术分队队形控制方法。

本文基于 Leader-Follower, 在由不规则三角网 (TIN) 组成的战场导航网格中采用了两阶段路径搜索方法, 首先在代价函数中结合线状地物的通行代价, 利用 A*算法寻找由三角形通道和可利用地物组成的路径; 然后使用改进的 Funnel 算法, 结合队形规模的约束对路径做平滑处理, 得到平滑的移动序列; 使用 morphing 技术^[7], 生成源队形到目标队形光滑的队形变化, 结合地形障碍约束, 修正并产生相应的中间约束队形序列, 再

基金项目: 国家自然科学基金面上项目 (61272199); 国家自然科学基金重点项目 (61532002)

作者简介: 陈逸 (1993-), 男, 上海人, 硕士研究生, 主要研究方向为计算机图形学 (416069384@qq.com); 高岩 (1973-), 男 (通信作者), 湖北襄阳人, 副教授, 博士, 主要研究方向为计算机图形学、图像处理; 王长波 (1976-), 男, 湖北随州人, 教授, 博士, 主要研究方向为计算机图形学。

使用提出的队形弹簧模型, 局部修正并控制 Follower 每一时刻的速度, 同时还加入了反馈机制, 动态调整 Leader 的速度大小以避免 Follower 掉队问题; 针对行进过程中动态障碍的避碰问题, 在相对速度障碍法 (RVO) 的基础上, 加入了速度协同控制以保持队形。最后通过仿真实验验证了方法的有效性。

1 算法原理

本文提出的基于 Leader-Follower 的改进队形控制算法, 在整体上将队形控制分为两阶段: Leader 寻径阶段和 Follower 跟随阶段。

常用的路径搜索算法有 A*算法、导航网格法、路径点法, 其中 A*算法为图搜索算法, 无法直接应用于三维战场; 路径点法需要在场景中标定路径点, 不适合复杂多变的战场环境; 而导航网格法因其贴合原有场景空间, 并且处理动态导航需求更加灵活, 目前被广泛应用, 但导航网格在跨越可穿行障碍物如壕沟时存在绕路的现象。因此在 Leader 寻径阶段采用两阶段寻径法, 结合 A*算法和 Funnel 算法产生平滑路径。

本文提出的基于 Leader-Follower 的改进分队队形控制寻径算法流程如图 1 所示。

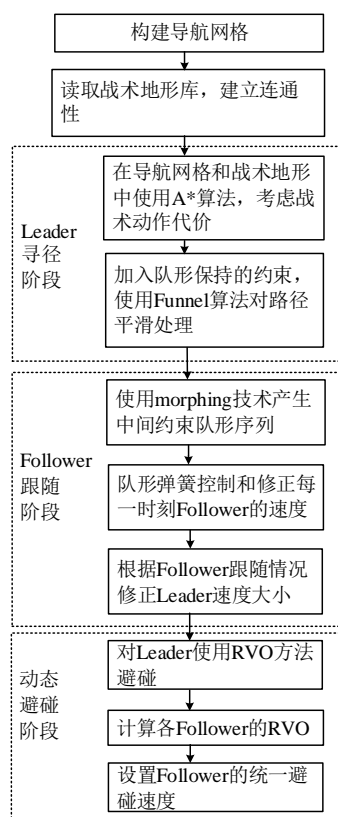


图 1 算法整体流程

2 Leader 寻径阶段

Leader 在三维场景中的寻径, 需要先构建战场对应的导航网格, 并在此基础上应用两阶段路径搜索算法。

2.1 构建导航网格

导航网格, 作为目前最常用的三维场景划分方法, 将三维

空间表示成由若干不规则三角形组成的网络。导航网格的构建分为五个主要步骤^[8], 过程如下:

- 对三维场景体素化, 将原始场景几何体抽象表示成实体高度场;
 - 生成导航区域, 通过检测实体高度场的上表面, 并将其划分成若干连通的区域, 同时通过判断区域是否过于接近障碍物以及是否有充足的空间来剔除无法行走的区域;
 - 检测导航区域的轮廓并将其表示成多边形, 并尽可能使外围边平滑;
 - 将轮廓多边形再分割成凸多边形;
 - 对凸多边形 Delaunay 三角化并加入高度信息。
- 构建出的战场导航网格如图 2 所示。

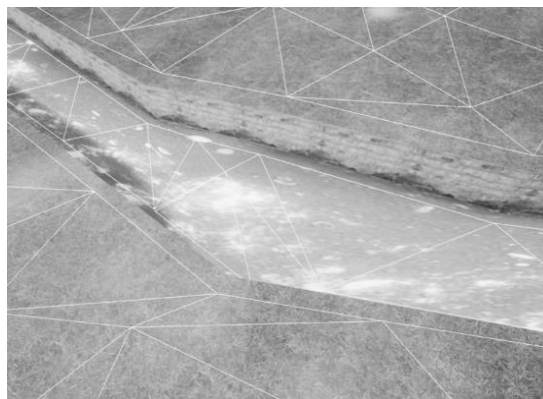


图 2 战场导航网格

2.2 两阶段路径搜索

在复杂的三维场景中, 只使用一种路径搜索算法往往难以使产生的路径同时满足最优和平滑, 特别在战场环境中, 存在各种可利用的地物, 如堑壕、掩体等, 士兵会执行攀爬、翻越等战术动作到达目的地。因此, 提出两阶段路径搜索法, 先在导航网格中使用 A*算法, 在考虑战术地形库附加的连通性和行为代价的基础上, 寻找导航网格中由三角形通道和可利用地物组成的路径; 再使用考虑队形约束的 Funnel 算法对三角形网格路径作平滑处理。

2.2.1 A*路径搜索算法

A*算法是经典的图路径搜索算法, 根据战场导航网格中三角形的连通性可以搜索起点到终点的三角形路径。为使士兵在寻径的过程中可以通过执行翻越动作, 利用地物到达目的地, 需要预先利用战术地形库建立三角形与地物的连通信息, 其中战术地形库存有战场中线状地物 (如堑壕、掩体等) 的位置信息以及执行战术动作穿越地物所花费的时间代价。

在 A*算法的搜索空间中, 除了考虑三角形间的连通性, 还需加入三角形与可利用地物的连通信息, 使 A*搜索到的路径还包括地物; 同时为了避免战术动作在局部不必要的情况下执行, 在 A*评估函数中, 状态 n 到目标状态最佳路径的代价估计函数 $h(n)$ 中除了需要考虑距离估计, 还要考虑翻越堑壕、掩体等线状地物所花费的额外花销。假设士兵执行动作所花费的时间

为 t_{action} , 士兵行走时的最大速率为 v_{max} , 则对应的花销为

$$\text{cost}(n) = t_{\text{action}} \cdot v_{\text{max}} \quad (1)$$

因此代价估计函数为

$$h(n) = \alpha \cdot \text{dist}(n) + \beta \cdot \text{cost}(n) \quad (2)$$

其中: α 为距离预计缩放参数; $\text{dist}(n)$ 为节点 n 到目标的欧氏距离; β 为地物开销预计缩放参数。

2.2.2 改进的 Funnel 算法

经 A* 算法计算而得到的是由三角形通道和地物组成的路径。为了在三维场景中获取平滑的路径, 首先需要计算路径中地物的对应战术动作执行位置, 然后计算各三角形通道内的最短路径。

Funnel 算法是在多边形内部寻找最短路径的一种高效算法, 时间复杂度为 $O(n)$, 其中 n 代表多边形三角化后其三角形的个数。Funnel 算法的思想是寻找多边形中的拐角点以形成自然的路径, 方法是通过设立漏斗 (图 3 (a) 中阴影处所示), 在遍历三角形的过程中不断缩小漏斗的开口角度 (图 3 (b)), 当漏斗闭合后则意味着找到了路径的拐角点, 并重新设置漏斗的位置 (图 3 (c)), 直到寻径结束。

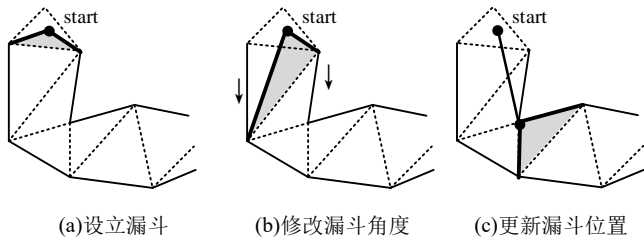


图 3 Funnel 算法计算拐角点过程

Funnel 算法在计算拐角点时并未考虑 agent 的大小, 直接应用在分队寻径上容易使士兵在拐角处被卡住, 因此需要在各拐角处, 计算分队的队形规模, 使分队在拐弯时有足够的空间。

对于由 A* 算法产生的路径 W_a , 使用满足队形规模约束的改进 Funnel 算法产生最终路径, 算法描述如下:

1) 遍历路径 W_a , 寻找其中包含的地物, 并以地物把 W_a 分割成局部三角形通道数组 C , 初始化最终路径 W 。

2) 对于 C 中每组相邻通道的邻近三角形 p_i 和 p_j , 找到 p_i 和 p_j 最近的两条边 e_i 和 e_j , 检测其与 R 中起点和终点构成的直线 e_{st} 是否相交, 如果相交则将交点作为动作执行点, 否则选择 e_i 和 e_j 与 e_{st} 距离最短的端点作为动作执行点。

3) 对于多边形通道 C_i , 起点为 s_i , 终点为 t_i , 将 s_i 加入 R , 连接 s_i 与 s_i 所在的多边形临边的左、右端点 n_l 和 n_r , 称为左线和右线。

4) 找到 C_i 下一个临边的左端点, 判断其是否在左线和右线组成的区域内, 如果在, 则 n_l 更新成此端点; 如果不在, 则维持 n_l 不变。同样的方法对右端点进行判断。

5) 重复步骤 4), 直到下一个左端点和右端点均在左线和右线组成的区域之外, 停止循环, n_l (或 n_r) 为拐角点, 并重复步骤 4), 直到找到所有的拐角点并保存至数组 T 。

6) 计算当前队形的最大外接圆半径为 r , 遍历拐角点数组 T , 在以各拐角点为圆心, 建立半径为 r 的圆, 计算相邻圆的公切线切点并加入 W , 最后再将 t_i 加入 R , 如图 4 所示。

重复执行步骤 3), 直到所有的通道都经过计算, 输出 R 为最终产生的路径。

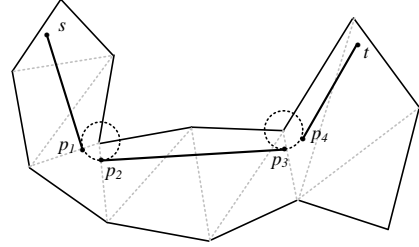


图 4 考虑队形约束的 Funnel 算法

3 Follower 跟随阶段

为使分队在作战仿真过程中保持队形, Leader-Follower 法通过控制 Follower 与 Leader 的距离 l 和方位角 ϕ 形成不同拓扑网络结构^[9], 而当分队队形需要在作战过程中改变时, 通常的做法是在预设的阵型中切换^[10], 但这样会使 Leader 与 Follower 在单一阵型中的位置相对固定, 难以自适应于战场地形环境, 在表现上不够逼真。

3.1 队形弹簧模型

在现实生活中, 弹簧在遇到障碍物时, 会呈现收缩和伸长两种状态以贴合障碍物, 而反映在分队的行进过程中, 队形同样需要具备动态贴合战场环境的能力。除此之外, 弹簧受到的弹力与其变化的长度成正比, 而当分队成员距离目标位置越远, 则其靠近的趋势, 即成员接近速度同样也应该越大。由此可以看出, 分队成员在队形保持时的移动与弹簧所反映出的特性基本一致, 因此提出在 Leader 和 Follower 间构建队形弹簧, 队形弹簧会受周遭环境障碍物挤压或在其他外力影响下动态伸缩, 使 Follower 在尽可能保持队形的基础上, 动态改变其与 Leader 的相对位置, 并且同时控制 Follower 的速度, 使其尽可能反映真实的情况。

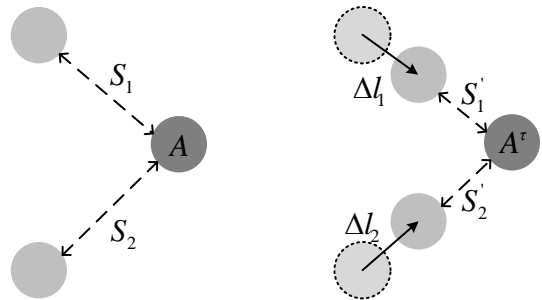


图 5 队形弹簧在 τ 时间内收缩相应长度

如图 5 所示, A 表示由三名成员组成的分队, 若经过 τ 时间, A' 的队形弹簧 S_1, S_2 分别受外力影响, 弹簧的长度改变了 Δl_1 和 Δl_2 , 形成新的队形弹簧, 则当分队中某个 Follower 的队形弹簧长度变化为 Δl 时, 根据胡克定律, 对应弹力 F_s 为

$$F_s = -k \cdot \Delta l \quad (3)$$

其中: k 为队形弹簧的劲度系数, 表示速度受弹簧形变长度的影响程度, 本文中 k 为常数, 取值为 $(0, 50]$ 。则 Follower 对应的加速度 a_F 为

$$a_F = \frac{F_s}{m} \quad (4)$$

其中: m 为 Follower 的质量, 默认 $m=1$, 则 Follower 与 Leader 在 t 时刻 ($t \in [0, \tau]$) 的相对速率为

$$v(t) = \left| \int_0^t a_F dx \right| = |t \cdot a_F| = t \cdot k \cdot \Delta l \quad (5)$$

为使 Follower 按对应队形移动, 除速率外, 还需要计算其对应的速度方向, 在 τ 时刻, Follower 对应的队形弹簧 S_i 的长度为 l_i 和 S_i 与 Leader 运动方向的夹角为 ϕ_i , 若 Leader 以 \mathbf{v}_0 的速度移动, 单位向量 $\mathbf{n}_0 = (x_0, y_0, z_0)$, 则 Follower 与 Leader 的期望相对位置为

$$E_i = (l_i \cdot n_{i,x} \quad l_i \cdot n_{i,y} \quad 0)^T \quad (6)$$

其中: \mathbf{n}_i 为 \mathbf{n}_0 经 ϕ_i 旋转后的方向向量, 则 Follower 在 t 时刻 ($t \in [0, \tau]$) 的速度方向为

$$\Psi_i(t) = \frac{E_i - P_i(t)}{\|E_i - P_i(t)\|} \quad (7)$$

其中: $P_i(t)$ 表示 t 时刻 Follower 在队形中与 Leader 的相对位置。

因队形弹簧会实时根据场景发生改变, 可能会导致的 Follower 掉队问题, 为解决这一问题, 引入了反馈机制, 动态调整 Leader 的速度, 以保证队形的稳定。 d_{\max} 表示 Follower 们的当前位置与其期望位置距离的最大值, $d_{\text{threshold}}$ 表示运行 Follower 与期望位置所允许的距离偏差的阈值。当 $d_{\max} > d_{\text{threshold}}$, 意味着 Follower 与其期望位置过远, 则调整 Leader 的速度为

$$\mathbf{v}_{\text{leader}} = \frac{\mu \cdot d_{\text{threshold}} \cdot \mathbf{v}_0}{d_{\max}} \quad (8)$$

其中: μ 是速度调节参数, $\mu \in [1, +\infty)$ 。

3.2 队形控制

在战场环境中, 分队具有多种战斗队形, 基本形态有三角形、梯形队形、横队队形、纵队队形, 士兵间最大间距为 20~30 m。

3.2.1 无障碍物约束的队形变换控制

本文采用 morphing 渐变技术控制产生平滑的队形过渡。Morphing 是指将源几何形状平滑地变换到目标形状, 并保持变换能量最少。假设当前队形为 A 阵形, 处于 C_1 位置; 目标队形为 B 阵形, 处于 C_2 位置, 阵形变换问题对应于二维多边形渐变问题, 使用二维形状 morphing 算法^[7]生成中间约束队形序列, 中间约束队形序列越多, 变换过程越光滑。算法步骤如下:

1) 计算源队形与目标队形各顶点到中心点的向量序列, 分别为 $S = \{S_i | i = 0, 1, \dots, n\}$ 和 $D = \{D_j | j = 0, 1, \dots, m\}$ 。

2) 建立源队形到目标队形各顶点的映射关系, 其中需要考虑 S_i 向 D_j 变换所扫过的面积和伸缩量, S_i 到 D_j 对应的代价函

数为 $\text{Cost}(S_i, D_j) = w_1 \times K(i, j) + w_2 \times L(i, j)$, 其中 $0 \leq w_1, w_2 \leq 1$,

$w_1 + w_2 = 1$ 。面积计算函数 $K(i, j) = \frac{1}{2}(|S_i| \times |D_j|) \times \sin(\theta_{ij})$, 其中 θ_{ij}

为 S_i 与 D_j 的夹角, 伸缩量计算函数 $L(i, j) = \|S_i\| - \|D_j\|$, 设源队形到目标队形的映射关系 $Z: \{S_i\} \rightarrow \{D_j\}$, 则其代价函数为

$$\text{Cost}(S, D, Z) = \sum_{i=0}^{m-1} \text{Cost}(S_i, D_{Z(i)}), \quad \text{代价函数的值越小则说明变换效果越好, 使用动态规划法计算最佳映射匹配。}$$

3) 按预设的时间步长, 根据顶点映射产生源阵型到目标阵型的中间约束队形序列。

3.2.2 有障碍约束的队形变换控制

战场环境存在各种类型的障碍物, 将导致缺省的目标阵形无法在目标位置被容纳的现象。此时, 需要对目标队形形状进行修正。本文采用贪心法探测目标位置处的最佳目标阵形几何形状, 算法过程描述如下:

- 1) 初始化旋转角度 θ 为 0, 缩放因子 S 为 1。
- 2) 判断目标队形形状各顶点位置是否与障碍物碰撞。如果没有, 则跳转 5); 否则继续执行 3)。
- 3) 将 θ 增加 3° , 对目标几何形状旋转 θ 角, 如果 $\theta < 360^\circ$, 则跳转 2); 否则继续执行 4)。
- 4) 将 S 乘以缩放系数 (本文设为 0.98), 对目标几何形状缩放 S 倍, 跳转 2)。
- 5) 该目标几何形状即所求的最佳目标几何形状。

算法的实质是对缺省目标阵形反复执行旋转和缩放操作, 在目标位置找到第一个能容纳下的经修正的目标队形, 对原队形和用修正后的目标队形调用 3.2.1 节的 morphing 技术即可产生有障碍约束下的中间约束队形序列。

值得注意的是, 上述最佳目标阵形几何形状探测算法只保证了目标位置处阵形的合法性, 中间约束队形序列仍然可能发生与障碍物冲突的情形, 本文提供两种策略予以解决。一种是对每一中间约束队形同样调用上述贪心算法检测出满足约束的修正几何形状, 再由队形弹簧模型控制角色移动; 另一种则允许分队自动切换队形, 当目标几何形状压缩率过大时 (一般在窄路或复杂障碍环境), 自动切换为一字队形, 以避免队形过度压缩的现象。

3.2.3 队形移动

在中间约束队形序列产生并修正后, Leader 开始沿路径移动, Follower 与 Leader 间的队形弹簧则在对应的时间内伸缩至中间约束队形相应的位置, 并按式(5)和 (7)产生每一时刻的速度。同时为避免 Follower 在移动过程被障碍物卡住, 分队需要每一时刻检测当前的队形弹簧是否受障碍物挤压, 如果有则在局部更新 Follower 的速度方向以避开障碍物。

4 动态障碍避碰

在作战仿真中, 三维场景中必然存在静态障碍和动态障碍, 而随着队形弹簧和导航网格的引入, 对于静态障碍已经可以产

生避免碰撞的平滑路径, 并且分队队形可以根据障碍物动态切换, 对于动态障碍采用考虑队形保持的相对速度障碍法^[11]。

假设存在两个 agent: A 和 B, 速度障碍 $VO_{A|B}^r$ 表示经过 τ 时间, A 和 B 发生碰撞的相对速度集合, 若将 agent 以圆表示, 则

$$D(P, R) = \{Q | \|Q - P\| < R\} \quad (9)$$

其中: P 和 R 表示 agent 圆的圆心和半径, 则速度障碍为

$$VO_{A|B}^r = \{v | tv \in D(P_B - P_A, R_A + R_B)\} \quad (10)$$

其中: $t \in [0, \tau]$, 则对应的相对速度障碍 $RVO_{A|B}^r$ 则为

$$RVO_{A|B}^r(v_A) = \{\dot{v}_A | 2\dot{v}_A - v_A \in VO_{A|B}^r\} \quad (11)$$

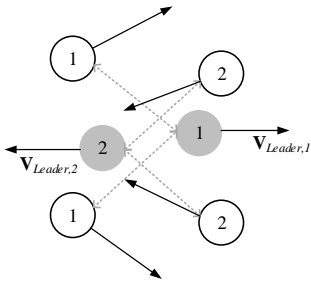


图6 分队队形因动态避碰而被冲散

在一般场景中, 只需要找到不在相对速度障碍范围的速度, 则就可以完成动态避碰, 但直接应用于分队的动态避碰会因分队内各 Follower 的避障方向选择不统一, 从而导致队形被冲散, 如图 6 所示。

为解决这一问题, 在相对速度障碍法的基础上加入分队成员速度协同控制, 通过设立 Follower 的统一避碰速度 v_{avoid} , 对分队内的成员分别计算在 τ 时间内的相对速度障碍 RVO^r , 使对于附近的动态障碍, 都有 $v_{avoid} \notin RVO^r$, 同时 v_{avoid} 要尽可能靠近原本的 v_{Leader} , 尽量使分队的避碰幅度小。

避碰速度计算过程如图 7 所示, 左边是分队 1 和分队 2 相遇的情况, 右边是分队 2 各成员所对应的相对速度障碍, 避碰速度 v_{avoid} 不能位于图中灰色虚线多边形内, 同时要尽可能接近原本的速度。

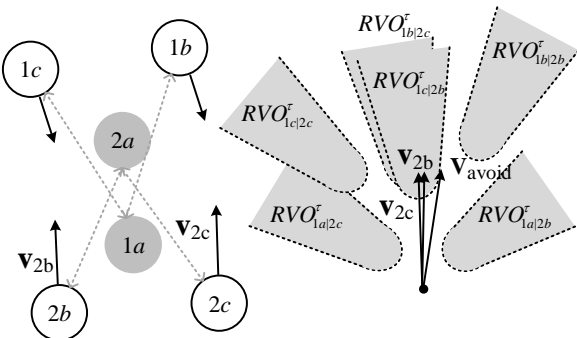


图7 动态障碍的避碰方向选择

在某些情况下, 发生碰撞是不可避免的, 比如两个分队在

狭窄的通道相遇, 此时双方都无法找到可行的避碰速度。因此当这类情况发生时, 选择冲突中的一方原地等待, 让另一方行动, 当冲突解除后再令等待的分队继续行动。

5 实验结果

本文实现了基于虚幻引擎 4 的作战仿真系统, 并使用了改进的分队队形控制算法, 分别对复杂障碍物环境下的分队队形变换控制和分队动态避碰进行仿真实验, 能在 PC 机 (i7-7700HQ、16 GB 内存、GTX 1060) 上以 60 帧每秒的速率运行。

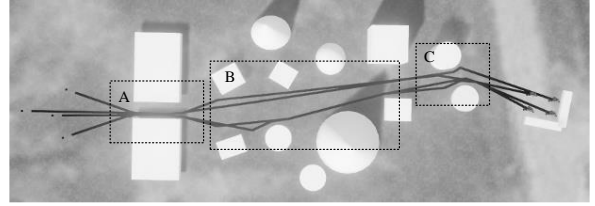
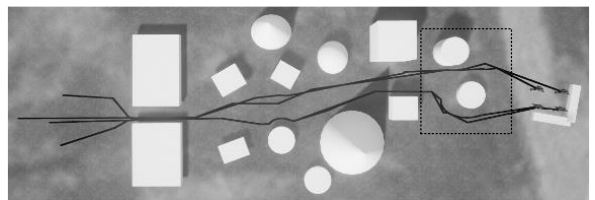


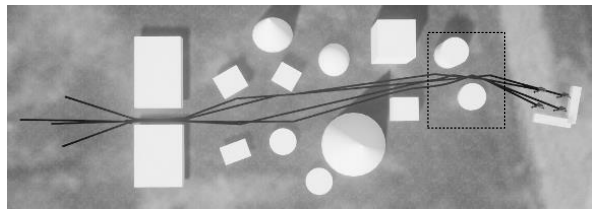
图8 队形变换效果

假设现有一四人小队, 初始以三角队形排列, 现令其机动至目标位置并变换至方阵, 在复杂障碍物场景下分队移动的效果如图 8 所示。当分队移动至区域 A, 通过判断地形无法容纳由 morphing 计算出的中间约束队形, 分队自动转换成一字纵队, 离开区域 A 后, 分队在区域 B 内变换到方阵队形并移动, 遇到较窄的区域 C 时队形压缩至合适位置, 并最终达到目标位置。从图中可以看出, 由 morphing 产生的中间约束队形, 通过队形弹簧控制每一时刻 Follower 的速度, 可以使分队控制的过程光滑合理。

本文还实现了其他队形控制方法, 分别为加入切线避障法的 Leader-Follower 法和基于人工势场的队形控制方法^[10], 应用在图 8 的场景中, 效果如图 9 所示。从图 9 (a) 的方框处可以看出, 人工势场法在分队躲避障碍物时会使得队形分散; 而从图 9 (b) 的方框处可以看出, 队形因障碍物限制而转换成了一路纵队, 但本文方法通过 3.2.2 节的修正算法缩小队形使分队仍保持方阵。



(a) 人工势场法



(b) 加入切线避障法的 Leader-Follower 法

图9 其他队形控制方法

假设现有两个分队相遇, 局部避碰的效果如图 10 所示。图 10 (a) 为两分队刚开始相遇的情况, 从图 10 (b) 和 (c) 可以看出, 分队在局部相遇避碰时依旧保持了各自的三角阵型, 没有被冲散。

(a) $t = 0s$ (b) $t = 1.2s$ (c) $t = 2.4s$ (d) $t = 3s$

图 10 分队相遇避碰

6 结束语

本文提出了基于 Leader-Follower 算法的改进队形控制方法, 可使分队在战场环境面对复杂场景仍能较好地保持队形。算法将分队控制分为 Leader 寻径阶段和 Follower 跟随阶段。在

Leader 寻径阶段中, 通过 A*算法和改进的 Funnel 算法产生考虑队形规模约束的平滑路径。在 Follower 跟随阶段, 采用 morphing 技术生成中间约束队形序列, 并通过 Leader 与 Follower 之间构建队形弹簧, 局部控制和修正每一时刻 Follower 的速度; 同时加入反馈机制动态修正 Leader 的速度以避免 Follower 产生掉队, 并基于相对速度障碍法及速度协同控制算法保证了队形在避碰过程的一致性。最后通过仿真实验验证了算法可行性。

参考文献:

- [1] 高强, 吕东浩, 庞毅. 多机器人队形保持与变换仿真研究 [J]. 自动化与仪表, 2012, 27 (10): 4-7, 52.
- [2] 翟红生, 王佳欣. 基于人工势场的机器人动态路径规划新方法 [J]. 重庆邮电大学学报: 自然科学版, 2015, 27 (6): 814-818.
- [3] Nair R R, Behera L, Kumar V, et al. Multisatellite formation control for remote sensing applications using artificial potential field and adaptive fuzzy sliding mode control [J]. IEEE Systems Journal, 2015, 9 (2): 508-518.
- [4] 李磊, 李小民, 郑忠贵, 等. 基于一致性理论的四旋翼无人机分布式编队控制方法 [J]. 电光与控制, 2015, 22 (10): 19-23, 29.
- [5] 王佳. 多 agent 系统的控制及稳定性分析 [D]. 南京: 南京理工大学, 2008.
- [6] 黄珏, 秦龙, 尹全军, 等. CGF 队形形成及避障行为研究 [J]. 计算机应用研究, 2012, 29 (12): 4466-4468.
- [7] 赵建明. 基于几何约束机制的群体队形控制方法研究 [D]. 合肥: 合肥工业大学, 2013.
- [8] Hale D H. A growth-based approach to the automatic generation of navigation meshes [D]. Charlotte: The University of North Carolina at Charlotte, 2011.
- [9] 袁学敏. 基于 Leader-Follower 理论的四旋翼无人机编队控制系统研究 [D]. 南京: 南京邮电大学, 2016.
- [10] 黄珏. CGF 的队形控制方法研究 [D]. 长沙: 国防科学技术大学, 2012.
- [11] Van Den Berg J, Lin M, Manocha D. Reciprocal velocity obstacles for real-time multi-agent navigation [C]// Proc of IEEE International Conference on Robotics and Automation. 2008: 1928-1935.